DAS ELEKTRONISCHE MAGAZIN 3/85

INPUT 64

Infos · News · Programme · Unterhaltung · Tips



MONITOR — sofort einsatzfähig
mit Mini-Assembler

Super-Sprite-Generator
Bewegte Grafik selbstgestaltet

Formel-Plotter
Mathematische Funktionen im Bild

Video-Kurs III —
diesmal alles über Sprites

Wettbewerb:
3000 Mark zu gewinnen!

Dokumentation und Bedienungshinweise

INPUT-ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 2746, 3000 Hannover 1.

Disketten Einstein DM 3,95
5 1/4 lochverstärkt, nach Lochung doppelseitig benutzbar (Abgabe nur im 25er Pack)

GP 700 VC DM 1.498,00
Seikosha Farbdrucker, grafikfähig, mit Interface f. C64

Diskettenaufbewahrungsbox DM 28,00
faßt 80 Stck. 5 1/4 Disketten, mit rauchfarbener Abdeckung.

Datenrecorder DM 89,00
neue Form, 6 Monate Garantie, für C64, VC20. Mit unserem Adapter C16 läßt sich der Datenrecorder auch an den C16, 116, + 4 anschließen. DM 17,80.

quick data drive DM 298,00
neues Speichermedium für C64/VC20. Geschwindigkeit wie eine Floppy aber nur halb so teuer. Einfaches Schnellkopieren von der Floppy bzw. Datenrecorder, möglich.

Floppy 1541 DM 628,00
hergestellt in Deutschland.

Commodore C 64 DM 559,00
(Abgabe nur in Verbindung mit Datenrecorder oder Floppy).

Digitalthermometer DM 45,00
(so groß wie ein normales Fieberthermometer) kein Quecksilber, schnelles Meßergebnis, Genauigkeit im Temperaturbereich von 34-44 °C = ± 1 °.

Akustikkoppler DM 398,00.
Dataphon 21 s, mit Interface und Software für C64, mit FTZ Nr.

MS 15 DM 1.498,00
Seikosha Typenraddrucker, mit Interface f. C64

Mail box DM 298,00
Modem mit Software und Anschluß an C64, ohne FTZ Nr., ein Leckerbissen für den Kenner.

Entstörzwischenstecker DM 24,80
verhindert weitgehend Programmabstürze wie sie oft durch Induktionsspannungen von Haushaltsgeräten hervorgerufen werden.

userport-Stecker DM 8,50
für C64

GP 100 VC DM 455,00
Seikosha Drucker, grafikfähig, mit Interface f. C64

Diskettenlocher DM 15,80
endlich lassen sich die Disketten problemlos beidseitig benutzen!

Wir haben eine eigene Spezialwerkstatt für Commodore und Apple Geräte.

Bitte schicken Sie Ihren Computer (auch wenn Sie ihn woanders gekauft haben), mit genauer Fehlerbeschreibung an uns ein. Wir garantieren eine zügige Bearbeitung.

Hochstrasse 1 · 588 Lüdenscheid

COMPUTER

gibt's bei

HAMPF electronic gmbh

Tel. (02351) 25358

Versand unfrei per Nachnahme, Prospekte gegen DM 1,30 in Briefmarken.

INHALTSVERZEICHNIS

INPUT 64

3/85

Liebe(r) 64er-Besitzer(in),

feed-back haben wir von Ihnen genug bekommen: INPUT 64 hat noch mehr Anklang gefunden, als wir in unseren kühnsten Träumen je erhofft hätten. Deswegen erscheint INPUT 64 ab Ausgabe 4/85 auch auf Diskette - also schon ab der nächsten Nummer! (16.80 DM plus 3.00 DM Porto und Versandkosten, nur im Direktbezug beim Verlag)

Die Rückmeldungen von uns an Sie dauern - aus produktionstechnischen Gründen - etwas länger. Die Rätselauflösung und die Gewinner des Wettbewerbs aus der ersten Ausgabe stellen wir Ihnen diesmal vor, ebenso letzte Berichtigungen zur ersten Nummer. Auf diesen Turnus können Sie sich einstellen: während Sie dieses Magazin in der Hand halten, ist der Redaktionsschluß für INPUT 64 Band 4/85 schon verstrichen.

Ansonsten: wie immer, viel Spaß!!

Ihre INPUT 64-Redaktion

1.) Entfernen Sie - bei ausgeschaltetem Rechner - evtl. vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.

2.) Legen Sie die erste Seite der Kassette ein und spulen Sie bis zum Bandanfang zurück.

3.) Geben Sie (LOAD) und (RETURN) ein - alles weitere geschieht von selbst. INPUT 64 meldet sich mit "INPUT 64" (womit auch sonst) und wünscht Ihnen anschließend "Guten Tag". Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.

4.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise "Bitte Band zurückspulen" undsoweiter ...

5.) Merken Sie sich vier Kommandos:

- (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.

- (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum "Band zurückspulen" aufgefordert.)

- (CTRL) und (s) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen. Auch hier bestätigen die Ausnahmen die Regel: Das "MLM 64- Demo" läßt dies nicht zu. Der Autor hat zu massiv in die Interruptroutine eingegriffen.

- (CTRL) und (s) macht Raubkopierer arbeitslos. Dieser Tastendruck steht für SAVEn und funktioniert bei allen Programmen, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwendungsprogramme etc.. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen die weiteren Anweisungen - Sie haben eine Kopie des Programms. Diese Option ist bei vielen Programmen nur am Programanfang möglich - entnehmen Sie dies jeweils den Hinweisen zu den einzelnen Modulen in diesem Heft.

- (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

6.) Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück.

Kassetten-Inhaltsverzeichnis Ausgabe 3/85

Auf der 2.Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte zunächst das Kapitel "Bei Ladeproblemen"!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner abstürzen. Gelangen Sie aber versehentlich in den Direktmodus, so befördert Sie ein SYS50307 wieder in das INPUT 64-Betriebssystem.

T E C H N I S C H E S

Daß Ihre Programme lauffähig und absturzsicher sind, versteht sich von selbst. Im Einzelnen heißt das: Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler abfangen, die Bildschirmmaske wird nicht zerstört undsoweiter.
Das Programm darf nur in C-64 BASIC oder in 6502/6510 Assembler geschrieben sein.
Alle Programme müssen auch ohne Floppy lauffähig sein, INPUT64 ist ein Kassettenmagazin. Floppy optional ist erlaubt und gewünscht.
Senden Sie uns Ihre Programme bitte auf Kassette oder Disk mit kommentiertem Listing und Kurzbeschreibung.
Wichtig: Sie haben die vollen Urheberrechte an Ihrem Programm und überlassen sie uns zur Erstveröffentlichung.

Außerdem gibt es einige, durch das INPUT64 Betriebssystem bedingte, programmiertechnische Erfordernisse:
1. Belegen Sie nur den Bereich des normalen BASIC RAM ($0800-$9FFF) und unter dem BASIC ROM ($A000-$BFFF).
2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystemvektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein.
3. Die Programme müssen als BASIC-File und mit .RUN. zu starten sein.
4. Die CTRL-Taste darf nicht benutzt werden.

Und geben Sie bitte auf Listings, Disks u.ä. den Programmnamen sowie Ihre Anschrift an.

M O B E D

Wenn Sie den Spritegenerator auf Ihren eigenen Datenträger abgespeichert und anschließend wieder in den Rechner geladen haben, starten Sie einfach mit RUN. Die genaue Bedienung entnehmen Sie bitte den anschließenden Bildschirmausdrucken.

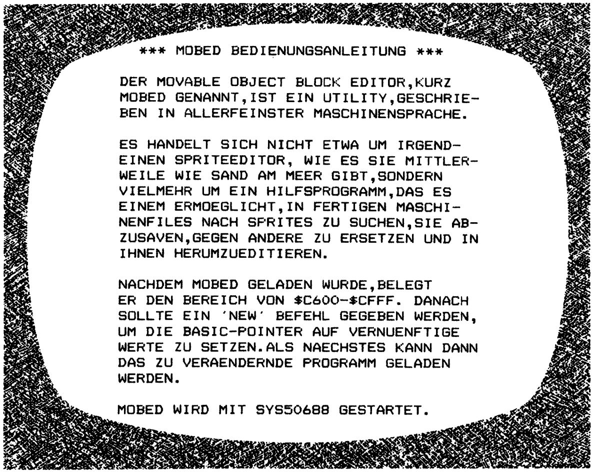
Innerhalb von INPUT 64 sind einige Funktionen nicht möglich:

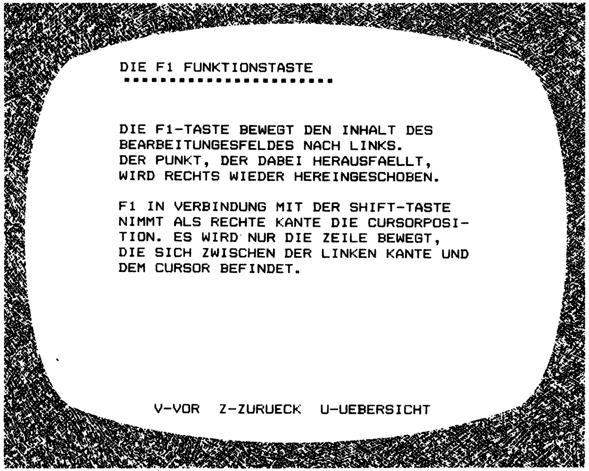
W = Write: Schreiben in den Speicher

K = Kassette: Laden von und Speichern auf Kassette der Sprites nur außerhalb des Magazins möglich!

D = Data: siehe "K"

HARDCOPYS: "MOBED"





DIE F3 FUNKTIONSTASTE

DIE F3-FUNKTION ARBEITET ANALOG ZUR F1-FUNKTION IN VERTIKALER RICHTUNG.

FUER DETAILLIERTE ERLAEUTERUNG EINE SEITE ZURUECKBLAETTERN.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE F5 FUNKTIONSTASTE

MIT DER F5-TASTE WIRD DER ADRESSZEIGER HOCHGEZAEHLT.

DER WERT DIESES ZEIGERS WIRD IN DER LINKEN OBEREN ECKE DES BEARBEITUNGSFELDES HEXADEZIMAL ANGEZEIGT (LOC:).

DRUECKEN DER F5 TASTE ERHOEHT DEN ZEIGER UM EINS. F5 MIT SHIFT ZUSAMMEN ZAEHLT 256 ($100) DAZU.

DIE FUNKTION DES ZEIGERS IST BEI DEN BEFEHLEN R UND W ERKLAERT.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE F7 FUNKTIONSTASTE

F7 IST IDENTISCH MIT F5. ALLERDINGS WIRD HIERBEI DER ADRESSZEIGER VERRINGERT.

FUER GENAUERE INFORMATIONEN EINE SEITE ZURUECKBLAETTERN.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE BRITISH POUND (£) TASTE

DIE £-TASTE DIENT ZUM AENDERN DER SPEICHERAUFTEILUNG DES RECHNERS. SIE FINDET IHRE ANWENDUNG IN VERBINDUNG MIT DER R-FUNKTION.

BETAETIGEN DIESER TASTE VERAENDERT DEN ZUSTAND DER MEMORY MANAGEMENT BITS IM 6510 I/O PORT. DADURCH IST ES MOEGLICH, DEN GESAMTEN RAM-SPEICHER NACH SPRITES ZU DURCHSUCHEN.

DER ZUSTAND DER BITS WIRD IN DER MITTE DER KOPFZEILE DES BEARBEITUNGSFELDES WIEDERGEGEBEN (MEM:).

V-VOR Z-ZURUECK U-UEBERSICHT

DIE TASTEN 1,2 UND 3

MIT DEN ZAHLENTASTEN 1,2 UND 3 WERDEN DIE DREI SPRITEFARBEN EINGESTELLT. EINMALIGES BETAETIGEN ZAEHLT DIE FARBE IN C64-INTERNER REIHENFOLGE HOCH.

DIE TASTEN 1 UND 2 BESTIMMEN DIE FARBEN EINES MULTI COLOR SPRITES UND SIND DESHALB AUCH NUR BEI EINEM SOLCHEN VON FUNKTION.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE R (READ) FUNKTION

BETAETIGEN DER R-TASTE LIEST EIN SPRITE IN DAS BEARBEITUNGSFELD.

DER ANFANG DES SPRITEMUSTERS WIRD DURCH DEN WERT DES ADRESSZEIGERS (LINKS OBEN) ANGEGEBEN.

SHIFT UND R SCHALTET DEN READ MODUS EIN. DAS BEDEUTET, DASS WAEHREND DES AENDERN DES ADRESSZEIGERS DAS SPRITEMUSTER SOFORT IN DAS BEARBEITUNGSFELD UEBERNOMMEN WIRD.

SPRITES ZUM EXPERIMENTIEREN BEFINDEN SICH IM RAM AB ADRESSE $3F00.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE T (TURN) FUNKTION

DER T-BEFEHL ERZEUGT EINE 90-GRAD DREHUNG DES BEARBEITUNGSFELDES. ES WIRD ALLERDINGS NUR EIN FELD VON 21 MAL 21 PUNKTEN AB LINKER UNTERER ECKE GEDREHT.

MOECHTE MAN DIE RECHTEN DREI SPALTEN MIT IN EINE DREHUNG EINBEZIEHEN, SO EMPFIEHLT ES SICH, DAS SPRITE MITTELS DER TASTE F1 IN DIE RICHTIGE POSITION ZU BRINGEN.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE Z (ZERO) FUNKTION

DER Z-BEFEHL LOESCHT DAS BEARBEITUNGSFELD.

Z UND DIE SHIFT TASTE LOESCHT NUR DIE ZEILE, IN DER SICH DER CURSOR BEFINDET.

Z UND DIE COMMODORE-TASTE LOESCHT DIE SPALTE AB CURSORPOSITION.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE P (POINT) FUNKTION

MIT DEM P-BEFEHL KANN MAN EINZELNE PUNKTE SETZEN ODER LOESCHEN.

P ALLEINE SETZT EINEN PUNKT.

P UND SHIFT LOESCHT EINEN PUNKT.

DER CURSOR WIRD HIERBEI NICHT VON DER STELLE BEWEGT.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE W (WRITE) FUNKTION

DIE W-FUNKTION ERMOEGLICHT ES, DEN INHALT DES BEARBEITUNGSFELDES IM RAM ABZULEGEN.

DIE ANFANGSADRESSE IST LINKS OBEN IM BEARBEITUNGSFELD ZU SEHEN (LOC:).

V-VOR Z-ZURUECK U-UEBERSICHT

DIE (PFEIL LINKS) FUNKTION

MIT DER TASTE WERDEN DIE BEIDEN SPRITE-MODI (MULTICOLOR/SINGLECOLOR) UMGESCHALTET.

DIE FARBENWAHL ERFOLGT MIT DEN TASTEN 1,2 UND 3.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE TASTEN U,H,J,N

MIT DEN TASTEN U,H,J,N IST ES MOEGLICH, LINIEN ZU ZEICHNEN.

DIE LINIE WIRD IN DER RICHTUNG GEZOGEN, DIE DER POSITION DER TASTE AUF DER TASTATUR ENTSPRICHT.
D.H.: H-LINKS J-RECHTS U-HOCH N-RUNTER.

IN VERBINDUNG MIT SHIFT WIRD DIE LINIE GEZEICHNET. MIT DER COMMODORE TASTE WIRD DIE LINIE GELOESCHT.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE S (SIZE) FUNKTION
....

MIT DER S-TASTE KANN DIE GROESSE DES SPRITES VERAENDERT WERDEN.

DIE VERAENDERUNG BEZIEHT SICH NUR AUF DAS ANSCHAUUNGSSPRITE RECHTS UNTEN. DAS SPRITEMUSTER BLEIBT UNVERAENDERT.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE I (INVERSE) FUNKTION
....

MIT DER I-TASTE IST ES MOEGLICH, DEN INHALT DES BEARBEITUNGSFELDES ZU INVERTIEREN.

DABEI WERDEN STERNCHEN IN PUNKTE UND PUNKTE IN STERNCHEN UMGEWANDELT.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE D (DATA) FUNKTION
....

DIE D-FUNKTION ERZEUGT DATAZEILEN.

DIE ZEILEN WERDEN ANS ENDE DES IM SPEICHER BEFINDLICHEN PROGRAMMES AN- GEHAENGT.

- ACHTUNG - DIESER BEFEHL FUNKTIONIERT NUR, WENN NACH DEM LADEN DES MOBED EIN 'NEW' ODER EIN RESET GEMACHT WURDE (NICHT MIT RUNSTOP/RESTORE VERWECHSELN).

ANDERNFALLS KANN ES ZU DERBEN HAENGERN KOMMEN.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE RUNSTOP-TASTE
....

MIT DER RUNSTOP-TASTE KANN MAN AUS DEM MOBED AUSSTEIGEN.

ES WERDEN DIE BASIC ANFANGSBEDINGUNGEN HERGESTELLT.

DER TASTATUR-REPEAT-MODUS BLEIBT ALLERDINGS EINGESCHALTET.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE C (COPY) FUNKTION

EINMALIGES DRUECKEN AUF DIE C-TASTE SCHALTET DEN COPY-MODUS EIN.

DER COPY-MODUS BEWIRKT, DASS BEI DREHUNGEN UND SPIEGELUNGEN DAS ALTE BILD ZUSAETZLICH ZU DEM NEUEN ERHALTEN BLEIBT.

NOCHMALIGES DRUECKEN DER C-TASTE SCHALTET DEN COPY-MODUS WIEDER AB.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE F (FILES) FUNKTION

NACH BETAETIGEN VON F ERSCHEINT DIE AUFFORDERUNG EINEN FILENAMEN EINZUGEBEN. DER NAME MUSS 6-STELLIG SEIN. ERLAUBT SIND BUCHSTABEN UND ZAHLEN. NACH EINGABE DES 6-TEN ZEICHENS WIRD DAS FILE IN DAS BEARBEITUNGSFELD GELADEN.

F MIT SHIFT SAVED DAS BEARBEITUNGSFELD AB.

F MIT COMMODORE-TASTE ZEIGT DAS INHALTSVERZEICHNIS AN (NUR DISK). DURCH DRUECKEN DER SPACE-TASTE KANN MAN DEN NAECHSTEN EINTRAG DES INHALTSVERZEICHNISSES SEHEN.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE M (MIRROR) FUNKTION

DIE M-TASTE ERLAUBT EIN SPIEGELN DES BEARBEITUNGSFELDES IN ZWEI ACHSEN.

M ALLEINE SPIEGELT UM DIE VERTIKALE MITTELLINIE.

M MIT SHIFT TUT DAS AN DER HORIZONTALEN.

V-VOR Z-ZURUECK U-UEBERSICHT

DIE K (KASSETTE) FUNKTION

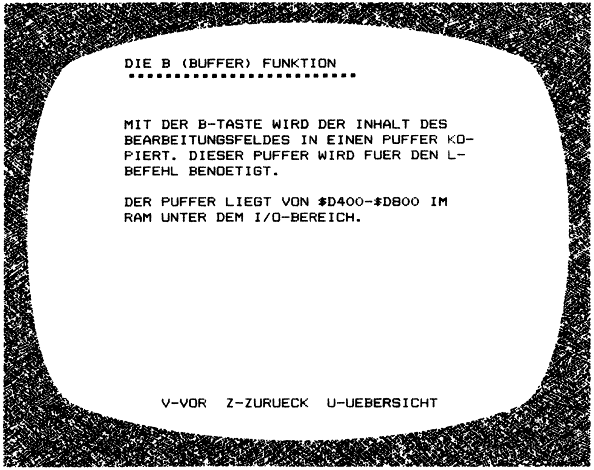
MIT HILFE DER K-TASTE KANN MAN DAS BEARBEITUNGSFELD AUF KASSETTE ABSPEICHERN ODER VON KASSETTE LADEN.

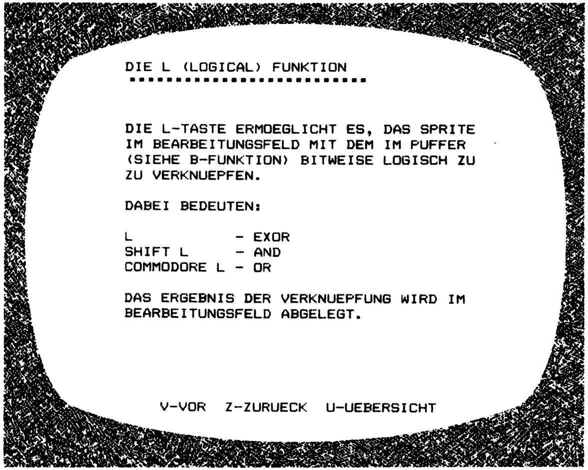
K LIEST VON KASSETTE;

SHIFT K SCHREIBT AUF KASSETTE.

MELDUNGEN WIE 'PRESS PLAY ON TAPE' WERDEN NICHT AUSGEGEBEN. DER RECORDER WIRD SOFORT NACH BETAETIGEN DER K-TASTE ANGESPROCHEN.

V-VOR Z-ZURUECK U-UEBERSICHT





Nachtrag:

Wenn Sie den MOBED innerhalb von INPUT 64 ausprobieren, schalten Sie doch einmal die "R"-Funktion (Read = Speicher auslesen) ein und sehen sich den Speicher genau an - mit den Tasten f5 und f7 durchblättern. Der Autor des Programms hat noch einige überraschungen hinterlegt.

H I L F S P R O G R A M M E

Zehn neue Funktionen stellen wir Ihnen diesmal vor, in der Hauptsache Sprite-Steuerungen.

Zunächst das Nebensächliche. Der Befehl "KALT" (Aufruf:Sys 49152) hat eine ähnliche Wirkung wie ein Reset-Taster - der Rechner wird in den Einschaltzustand versetzt, der Hauptspeicher allerdings nicht gelöscht. Nachteil: Hat sich der Computer wirklich "aufgehängt", können Sie diesen Befehl nicht mehr eingeben.

Ernstzunehmender sind die Befehle "PRINT AT" und der Aufruf zum Auslesen des Floppy-Fehlerkanals. Die PRINT-AT-Routine wird durch ein "SYS 49191" aufgerufen und erwartet die Angabe vom Zeile, Spalte und Text:

```
SYS 49191,11,7,"Tabelle"
```

setzt das Wort "Tabelle" in die 11. Zeile und die 7. Spalte. Als Parameter können sowohl für Zeile und Spalte als auch für den Text Variablen oder Ausdrücke übergeben werden. Dies gilt ebenso für sämtliche anderen Befehle. Illegale Werte oder fehlende Parameter werden durch die Commodore-üblichen Fehlermeldungen quittiert.

Die Ursache des Blinkens der roten LED an der Diskettenstation erfahren Sie durch ein "SYS 49223,ga" (ga = Geräteadresse, in der Regel 8).

Alles weitere dreht sich um Sprites.

Die folgenden sieben Befehle ersparen umständliches "POKEN" und komplizierte Berechnungen.

Allgemeine Syntax der Spritebefehle ist

```
SYS ADRESSE,SPRITENUMMER,PARAMETER 1,PARAMETER 2, ...
```

Sprite einschalten: SYS 49360,sn,1
Sprite ausschalten: SYS 49360,sn,0
(sn = Spritenummer)

Sprite-Block definieren: SYS 49328,sn,bn
(bn = Blocknummer)

Sprite-Farbe setzen: SYS 49654,sn,c
(c = Farbe)

Sprite-Multicolorfarben: SYS 49588,sn,c1,c2,c3
(c1-c3 =Farben)

Sprite-Größe und Priorität: SYS 49445,sn,x,y,p
(x = X-Vergrößerung, y = Y-Ver-größerung, p = Text- oder Spritepriorität) So ist nach "sys 49445,0,1,0,1" Sprite Nummer 0 in X-Richtung vergrössert, in Y-Richtung nicht und erscheint

hinter dem Text.

Sprite setzen: SYS 49399,sn,x,y
(x = X-Position, y = Y-Position, y darf nicht größer als 255 werden.)

Sprite bewegen: SYS 49638,sn,x1,y1,x2,y2,t
(t = Geschwindigkeit) Bewegung von Ausgangsposition x1/y1 nach Zielposition x2/y2. t=0 ist die schnellste Geschwindigkeit.

Die Tools werden aus dem Magazin heraus als Basiclader abgespeichert. (Häufig klappt dies erst beim zweiten Versuch, dies hängt zusammen mit der Unverträglichkeit von Sprites und Peripherieoperationen) Beim Neustart durch RUN wird das Maschinenprogramm in seinen eigentlichen Arbeitsbereich transferiert (von $C000 - $C3A0, außerhalb des Basic-Speichers.) Wenn Sie die dann folgende Abfrage mit JA beantworten, können Sie diese Maschinenspracheroutinen noch einmal gesondert abspeichern. Diese sind "absolut" wieder zu Laden, mit

```
LOAD " INPUT 64 MOB ",1,1
```

(bzw. ...,8,1 bei Diskettenlaufwerken)

Der Basiclader enthält außerdem mehrere Zeilen mit Variablenzuweisungen für die Aufrufadressen. Diese Variablenzuweisungen müssen natürlich gegebenenfalls durch ein RUN neu initialisiert werden! Sie können dann statt des "SYS 49360,0,1" zum Einschalten eines Sprites "SYS mo,0,1" eingeben.

Befehle	Variablenname	Aufrufadresse
Kalt-Start	ka	49152
Print At	pr	49191
Disk Error	de	49223
Mob Definition	md	49328
Mob on/off	mo	49360
Mob move	mm	49399
Mob-Größe	mg	49445
Mob-Colour	mc	49564
Mob-Farben	mf	49588
Mob transfer	mt	49638

Der Begriff "Mob" hat hier nichts mit Gesellschaftspolitik zu tun, sondern ist die, meines Erachtens treffendere, Bezeichnung für Sprites: Movable Object Block, zu deutsch Bewegliche Objekte.

ASSEMBLER-LISTING: "HILFSPROGRAMME"

2

```
profi-ass 64 v2.0    seite 1

25:     c000                        *=    $c000
26:                      ; tools input 64 3.85 src
30:     c000                        .opt p4,oo
50:     aefd             chkom     =     $aefd
60:     b248             illqu     =     $b248
70:     0073             charget   =     $0073
80:     0079             chargot   =     $0079
90:     d000             vic       =     53248
100:    b79e             getbyt    =     $b79e
110:    b7eb             getcor    =     $b7eb
120:    aaa0             print     =     $aaa0
130:    ffd2             bsout     =     $ffd2
140:    e510             csys      =     58640
150:    00d6             zeile     =     214
160:    00d3             spalte    =     211
170:    ab1e             strout    =     $ab1e
180:    ffe4             get       =     $ffe4
190:    ffba             setpar    =     $ffba
200:    ffbd             setnam    =     $ffbd
210:    ffc0             open      =     $ffc0
220:    ffc3             close     =     $ffc3
230:    ffcc             clrch     =     $ffcc
240:    ffc6             chkin     =     $ffc6
250:    ffcf             basin     =     $ffcf
260:    a437             error     =     $a437
1000:                    ;-------------------------------
1010:                    ; kalt = reset mit sicherheitsabf.
1020:                    ;
1030:   c000 a9 14       kalt      lda   #<katx
1040:   c002 a0 c0                 ldy   #>katx
1050:   c004 20 1e ab              jsr   strout    ; abfrage
1060:   c007 20 e4 ff    k1        jsr   get
1070:   c00a f0 fb                 beq   k1
1080:   c00c c9 4a                 cmp   #"j"
1090:   c00e d0 03                 bne   kout      ; nicht ja, dann ende
1100:   c010 6c fc ff              jmp   ($fffc)   ; reset einsprung
1110:   c013 60          kout      rts
1200:   c014 0d 0d       katx      .byte13,13
1210:   c016 12 53 49              .asc "█sicher ?? (j) █"
1220:   c026 00                    .byte00
1230:                    ;------------------------------
1240:                    ; prat = print-at routine
1245:                    ; syntax prat,zeile,spalte,"text"
1250:                    ;
1260:   c027 20 fd ae    prat      jsr   chkom
1270:   c02a 20 eb b7              jsr   getcor
1275:   c02d e0 28                 cpx   #40
1275:   c02f b0 13                 bcs   zugross
1280:   c031 a5 14                 lda   $14
1290:   c033 c9 19                 cmp   #25
1300:   c035 b0 0d                 bcs   zugross
1310:   c037 86 d3                 stx   spalte
1320:   c039 85 d6                 sta   zeile
1330:   c03b 20 10 e5              jsr   csys
1340:   c03e 20 fd ae              jsr   chkom
1350:   c041 4c a0 aa              jmp   print
1360:   c044 4c 48 b2    zugross   jmp   illqu
1370:                    ;
```

```
profi-ass 64 v2.0    seite 2

1380:                    ;--------------------------------
1390:                    ; derr = fehlerkanal auslesen
1400:                    ; syntax derr,ga
1410:                    ;
1420:   c047 20 fd ae    derr      jsr   chkom
1430:   c04a 20 9e b7              jsr   getbyt
1440:   c04d e0 05                 cpx   #05
1450:   c04f 90 37                 bcc   illdv
1450:   c051 e0 10                 cpx   #16
1450:   c053 b0 33                 bcs   illdv
1460:   c055 8a                    txa
1460:   c056 85 02                 sta   02
1465:   c058 a0 0f                 ldy   #15
1465:   c05a 20 ba ff              jsr   setpar
1470:   c05d a9 00       d1        lda   #00
1470:   c05f 20 bd ff              jsr   setnam
1480:   c062 20 c0 ff              jsr   open
1480:   c065 a5 02                 lda   02
1480:   c067 20 c3 ff              jsr   close
1480:   c06a 24 90                 bit   144
1480:   c06c 30 1f                 bmi   nodv
1485:   c06e e6 98                 inc   152
1490:   c070 a6 02                 ldx   02
1490:   c072 20 c6 ff              jsr   chkin
1500:   c075 20 cf ff    cch1      jsr   basin
1510:   c078 20 d2 ff              jsr   bsout
1520:   c07b 24 90       nb        bit   144
1520:   c07d 50 f6                 bvc   cch1
1530:   c07f a5 02       derrend   lda   02
1530:   c081 20 c3 ff              jsr   close
1540:   c084 20 cc ff              jsr   clrch
1560:   c087 60                    rts
1570:   c088 a2 09       illdv     ldx   #09
1570:   c08a 4c 37 a4              jmp   error
1580:   c08d a5 02       nodv      lda   02
1580:   c08f 20 c3 ff              jsr   close
1580:   c092 a2 05                 ldx   #05
1580:   c094 4c 37 a4              jmp   error
1590:                    ;--------------------------------


1600:                    ;
1610:                    ; ab hier mob-tools
1620:                    ;
1630:                    ;--------------------------------
1640:                    ;
1650:                    ; zero-page init ($a4,$a5), mobnummerpruefung
1660:                    ;


1670:   c097 a9 d0       spini     lda   #$d0
1670:   c099 85 a4                 sta   $a4
1680:   c09b a9 00                 lda   #$00
1680:   c09d 85 a3                 sta   $a3
1682:   c09f 20 fd ae              jsr   chkom
1682:   c0a2 20 9e b7              jsr   getbyt
1682:   c0a5 e0 08                 cpx   #08
1682:   c0a7 b0 24                 bcs   errsp1
1684:   c0a9 86 02                 stx   02
1684:   c0ab 20 fd ae              jsr   chkom
1684:   c0ae 60                    rts
1690:   c0af 60                    rts
```

```
1700:                  ;
1710:                  ;---------------------------------
1720:                  ;
1730: c0b0 20 97 c0 md        jsr  spini
1790: c0b3 20 9e b7           jsr  getbyt
1800: c0b6 8a                 txa
1800: c0b7 48                 pha            ; blockzeiger retten
1810: c0b8 ad 88 02           lda  648       ; high-byte vid-ram
1820: c0bb 18                 clc
1820: c0bc 69 03              adc  #03
1820: c0be 85 a4              sta  $a4
1830: c0c0 18                 clc
1830: c0c1 a5 02              lda  02
1830: c0c3 69 f8              adc  #$f8
1830: c0c5 85 a3              sta  $a3
1870: c0c7 68                 pla
1870: c0c8 a0 00              ldy  #00
1870: c0ca 91 a3              sta  ($a3),y  ; blockzeiger in spritezeiger
1875: c0cc 60                 rts
1876:                  ;
1877:                  ;---------------------------------
1878:                  ;

1880: c0cd 4c 48 b2 errsp1    jmp  illqu
1890:                  ;
1900:                  ;---------------------------------
1910:                  ;
1920:                  ;mo mob ein/aus
1930:                  ;

1940: c0d0 20 97 c0 mo        jsr  spini
1990: c0d3 20 9e b7           jsr  getbyt
2000: c0d6 e0 02              cpx  #02
2000: c0d8 b0 f3              bcs  errsp1
2010: c0da ca                 dex
2010: c0db d0 0c              bne  aus
2020: c0dd a6 02     ein      ldx  02
2030: c0df bd de c1           lda  bittab,x
2040: c0e2 0d 15 d0           ora  vic+21
2050: c0e5 8d 15 d0           sta  vic+21
2060: c0e8 60                 rts
2070: c0e9 a6 02     aus      ldx  02
2070: c0eb bd de c1           lda  bittab,x
2080: c0ee 49 ff              eor  #$ff
2090: c0f0 2d 15 d0           and  vic+21
2100: c0f3 8d 15 d0           sta  vic+21
2110: c0f6 60                 rts
2120:                  ;
2130:                  ;---------------------------------
2140:                  ;
2150:                  ; mm  mobmove
2160:                  ;

2170: c0f7 20 97 c0 mm        jsr  spini
2190: c0fa 20 eb b7           jsr  getcor
2210: c0fd 06 02              asl  02
2210: c0ff a4 02              ldy  02
2220: c101 c8                 iny
2220: c102 8a                 txa
2220: c103 91 a3              sta  ($a3),y
2230: c105 88                 dey
```



```
2230: c106 a5 14              lda  $14
2230: c108 91 a3              sta  ($a3),y
2240: c10a 46 02              lsr  02
2240: c10c a6 02              ldx  02
2250: c10e bd de c1           lda  bittab,x
2260: c111 a4 15              ldy  $15
2260: c113 f0 07              beq  mlow
2270: c115 0d 10 d0           ora  vic+16
2280: c118 8d 10 d0           sta  vic+16
2290: c11b 60                 rts
2300: c11c 49 ff     mlow     eor  #$ff
2310: c11e 2d 10 d0           and  vic+16
2320: c121 8d 10 d0           sta  vic+16
2330: c124 60                 rts

2340:                  ;
2350:                  ;---------------------------------
2360:                  ;
2370:                  ; mg sprite-groesse & prioritaet
2380:                  ;

2390: c125 20 97 c0 mg        jsr  spini
2400: c128 20 eb b7           jsr  getcor
2410: c12b a5 14              lda  $14
2410: c12d c9 02              cmp  #02
2410: c12f b0 68              bcs  errsp2
2420: c131 a4 15              ldy  $15
2420: c133 d0 64              bne  errsp2
2430: c135 e0 02              cpx  #02
2430: c137 b0 60              bcs  errsp2
2440: c139 e0 00              cpx  #00
2440: c13b f0 0d              beq  noy
2450: c13d a6 02              ldx  02
2450: c13f bd de c1           lda  bittab,x
2460: c142 0d 17 d0           ora  vic+23
2470: c145 8d 17 d0           sta  vic+23
2480: c148 d0 0d              bne  xgr
2490: c14a a6 02     noy      ldx  02
2490: c14c bd de c1           lda  bittab,x
2500: c14f 49 ff              eor  #$ff
2510: c151 2d 17 d0           and  vic+23
2520: c154 8d 17 d0           sta  vic+23
2530: c157 a5 14     xgr      lda  $14
2530: c159 f0 0b              beq  nox
2540: c15b bd de c1           lda  bittab,x
2550: c15e 0d 1d d0           ora  vic+29
2560: c161 8d 1d d0           sta  vic+29
2570: c164 d0 0b              bne  pr
2580: c166 bd de c1 nox       lda  bittab,x
2590: c169 49 ff              eor  #$ff
2600: c16b 2d 1d d0           and  vic+29
2610: c16e 8d 1d d0           sta  vic+29
2620: c171 20 fd ae pr        jsr  chkom
2630: c174 20 9e b7           jsr  getbyt
2630: c177 e0 02              cpx  #02
2630: c179 b0 1e              bcs  errsp2
2640: c17b e0 00              cpx  #00
2640: c17d f0 0c              beq  nopr
2650: c17f a6 02              ldx  02
2650: c181 bd de c1           lda  bittab,x
```

```
profi-ass 64 v2.0    seite 5

2660:  c184 0d 1b d0          ora  vic+27
2670:  c187 8d 1b d0          sta  vic+27
2680:  c18a 60                rts
2690:  c18b a6 02    nopr     ldx  02
2690:  c18d bd de c1          lda  bittab,x
2700:  c190 49 ff             eor  #$ff
2710:  c192 2d 1b d0          and  vic+27
2720:  c195 8d 1b d0          sta  vic+27
2730:  c198 60                rts
2740:                ;
2750:                ;---------------------------------
2756:  c199 4c 48 b2 errsp2   jmp  illqu
2757:                ;---------------------------------
2760:                ;
2770:                ; mc mob-colour
2780:  c19c 20 97 c0 mc       jsr  spini
2790:  c19f 20 9e b7          jsr  getbyt
2800:  c1a2 a4 02             ldy  02
2800:  c1a4 b9 de c1          lda  bittab,y
2800:  c1a7 49 ff             eor  #$ff
2800:  c1a9 2d 1c d0          and  vic+28
2800:  c1ac 8d 1c d0          sta  vic+28
2810:  c1af 8a                txa
2810:  c1b0 99 27 d0          sta  vic+39,y
2820:  c1b3 60                rts
2830:                ;
2840:                ;----------------------------------
2850:                ;
2860:                ; mf mob-multicolour
2870:  c1b4 20 97 c0 mf       jsr  spini
2880:  c1b7 a6 02             ldx  02
2880:  c1b9 bd de c1          lda  bittab,x
2890:  c1bc 0d 1c d0          ora  vic+28
2890:  c1bf 8d 1c d0          sta  vic+28
2900:  c1c2 20 eb b7          jsr  getcor
2910:  c1c5 a5 15             lda  $15
```

```
2910:  c1c7 d0 d0             bne  errsp2
2920:  c1c9 a5 14             lda  $14
2920:  c1cb 8d 25 d0          sta  vic+37
2930:  c1ce 8e 26 d0          stx  vic+38
2940:  c1d1 20 fd ae          jsr  chkom
2940:  c1d4 20 9e b7          jsr  getbyt
2940:  c1d7 8a                txa
2950:  c1d8 a4 02             ldy  02
2950:  c1da 99 27 d0          sta  vic+39,y
2950:  c1dd 60                rts
2960:                ;
2970:                ;--------------------------------
2980:                ;
2990:                ;
50000: c1de 01 02 04 bittab   .byte1,2,4,8,16,32,64,128
51000: c1e6          mt       =    *
61230: c1e6                   .sym
61240: c1e6                   .sst 8,2,"@:mob-beftab,s,w"
]c000-c1e6
no errors

profi-ass 64 v2.0    seite 6
symboltable:
mt       c1e6   bittab   c1de   mf       c1b4   mc       c19c
errsp2   c199   nopr     c18b   pr       c171   nox      c166
xgr      c157   noy      c14a   mg       c125   mlow     c11c
mm       c0f7   aus      c0e9   ein      c0dd   mo       c0d0
errspl   c0cd   md       c0b0   spini    c097   nodv     c08d
illdv    c088   derrend  c07f   nb       c07b   cch1     c075
d1       c05d   derr     c047   zugross  c044   prat     c027
katx     c014   kout     c013   k1       c007   kalt     c000
error    a437   basin    ffcf   chkin    ffc6   clrch    ffcc
close    ffc3   open     ffc0   setnam   ffbd   setpar   ffba
get      ffe4   strout   able   spalte   00d3   zeile    00d6
csys     e510   bsout    ffd2   print    aaa0   getcor   b7eb
getbyt   b79e   vic      d000   chargot  0079   charget  0073
illqu    b248   chkom    aefd
54 symbols defined
```

profi-ass 64 v2.0   seite 1

```
30:   c1e6                      .lst 8,3,"mob-beftab"
40:   c1e6                      .opt p4,oo
41:   00a6          y1          =    $a6
42:   00a7          x1l         =    $a7
43:   00a8          x1h         =    $a8
44:   00a9          y2          =    $a9
45:   00b4          time        =    $b4
46:   00b5          dy          =    $b5
47:   00b8          x2l         =    $b8
48:   00b9          x2h         =    $b9
49:   00b6          dxl         =    $b6
50:   00b7          dxh         =    $b7
51:   0014          xal         =    $14
52:   0015          xah         =    $15
53:   00bd          dsum        =    $bd
54:   00be          swl         =    $be
55:   00bf          swh         =    $bf
56:   0094          gx          =    $94
57:   0095          gy          =    $95
58:   0096          dsumh       =    $96
60:                 ; source sprite transfer
70:                 ; befehl mt
80:                 ; syntax sys mt,sn,x1,y1,x2,y2,t
90:                 ;---------------------------------


99:   c1e6                      *=   mt        ; 52706d , cde2h
100:  c1e6 20 f7 c0             jsr  mm        ; x1-cor in $14/$15
105:  c1e9 06 02                asl  02
110:  c1eb a6 02                ldx  02        ; spritenr. mal 2
115:  c1ed 46 02                lsr  02        ; spritenr. wiederherstellen
120:  c1ef e8                   inx            ; +1
130:  c1f0 bd 00 d0             lda  vic,x     ; y1-cor
140:  c1f3 85 a6                sta  y1        ; retten
150:  c1f5 a5 14                lda  $14
150:  c1f7 85 a7                sta  x1l       ; x1-cor retten
160:  c1f9 a5 15                lda  $15
160:  c1fb 85 a8                sta  x1h
170:  c1fd 20 fd ae             jsr  chkom
180:  c200 20 eb b7             jsr  getcor    ; x2-cor in $14/$15
190:  c203 86 a9                stx  y2
192:  c205 a5 14                lda  $14
192:  c207 85 b8                sta  x2l
192:  c209 a5 15                lda  $15
192:  c20b 85 b9                sta  x2h       ; x2-cor setzen
195:  c20d a5 a7                lda  x1l
195:  c20f 85 14                sta  xal
195:  c211 a5 a8                lda  x1h
195:  c213 85 15                sta  xah       ; akt. position = x1
200:  c215 20 fd ae             jsr  chkom
210:  c218 20 9e b7             jsr  getbyt
220:  c21b 86 b4                stx  time
220:  c21d a9 00                lda  #00
220:  c21f 85 94                sta  gx
220:  c221 85 95                sta  gy


230:                ;--------------------------------
240:                ; xdiff & ydiff bilden und vergleichen
250:                ;
260:  c223 38                   sec
270:  c224 a5 a9                lda  y2
```

profi-ass 64 v2.0   seite 2

```
270:  c226 e5 a6                sbc  y1
270:  c228 85 b5                sta  dy
280:  c22a b0 03                bcs  noinvy
280:  c22c 20 55 c3             jsr  ydinv
300:  c22f 38          noinvy   sec
300:  c230 a5 b8                lda  x2l
300:  c232 e5 a7                sbc  x1l
300:  c234 85 b6                sta  dxl
310:  c236 a5 b9                lda  x2h
310:  c238 e5 a8                sbc  x1h
310:  c23a 85 b7                sta  dxh
320:  c23c b0 03                bcs  noinvx
320:  c23e 20 3e c3             jsr  xdinv
330:  c241 38          noinvx   sec
330:  c242 a5 b6                lda  dxl
340:  c244 c5 b5                cmp  dy
340:  c246 b0 07                bcs  xdbig
350:  c248 a5 b7                lda  dxh
350:  c24a d0 03                bne  xdbig
360:  c24c 4c 84 c2             jmp  ydbig
370:                   ;
390:                   ;
400:                   ;----------------------------------
410:                   ; xdiff > ydiff


420:  c24f 18          xdbig    clc
420:  c250 a5 b6                lda  dxl
420:  c252 85 bd                sta  dsum
420:  c254 85 be                sta  swl
425:  c256 a5 b7                lda  dxh
425:  c258 85 bf                sta  swh
425:  c25a 85 96                sta  dsumh     ; sw=dx
427:  c25c 18                   clc
427:  c25d 66 96                ror  dsumh
427:  c25f 66 bd                ror  dsum      ; dsum=int(dx/2)
428:  c261 4c 7b c2             jmp  eqtest
430:  c264 20 24 c3    xloop    jsr  delay
430:  c267 20 b8 c2             jsr  dsdyad
440:  c26a 20 10 c3             jsr  dsdxcp
450:  c26d 90 06                bcc  xnoy
450:  c26f 20 ec c2             jsr  ystep
450:  c272 20 d2 c2             jsr  dsdxsb
460:  c275 20 f8 c2    xnoy     jsr  xstep
460:  c278 20 04 c3             jsr  swl
470:  c27b a5 be       eqtest   lda  swl
470:  c27d d0 e5                bne  xloop
480:  c27f a5 bf                lda  swh
480:  c281 d0 e1                bne  xloop
490:  c283 60          out      rts
495:                   ;
500:                   ;----------------------------------
510:                   ; ydiff > xdiff


520:  c284 18          ydbig    clc
520:  c285 a5 b5                lda  dy
520:  c287 85 bd                sta  dsum
520:  c289 85 be                sta  swl
520:  c28b f0 2a                beq  outy
530:  c28d a9 00                lda  #00
530:  c28f 85 bf                sta  swh
```

```
profi-ass 64 v2.0    seite 3

530:   c291 85 96            sta  dsumh    ; sw=dy
535:   c293 18               clc
535:   c294 66 96            ror  dsumh
535:   c296 66 bd            ror  dsum
540:   c298 20 24 c3 yloop   jsr  delay
540:   c29b 20 c4 c2         jsr  dsdxad
550:   c29e 20 1a c3         jsr  dsdycp
560:   c2a1 90 06            bcc  ynox
560:   c2a3 20 f8 c2         jsr  xstep
560:   c2a6 20 e0 c2         jsr  dsdysb
570:   c2a9 20 ec c2 ynox    jsr  ystep
570:   c2ac 20 04 c3         jsr  sw1
580:   c2af a5 be            lda  swl
580:   c2b1 d0 e5            bne  yloop
585:   c2b3 a5 bf            lda  swh
585:   c2b5 d0 e1            bne  yloop
590:   c2b7 60       outy    rts
595:                 ;
600:                 ;-------------------------------
790:                 ;
800:                 ;---------------------------------
810:                 ; dsum = dsum + dy

820:   c2b8 18       dsdyad  clc
820:   c2b9 a5 bd            lda  dsum
820:   c2bb 65 b5            adc  dy
820:   c2bd 85 bd            sta  dsum
830:   c2bf 90 02            bcc  dsdyad1
830:   c2c1 e6 96            inc  dsumh
850:   c2c3 60       dsdyad1 rts
860:                 ;---------------------------------
900:                 ;dsum = dsum + dx

910:   c2c4 18       dsdxad  clc
910:   c2c5 a5 bd            lda  dsum
910:   c2c7 65 b6            adc  dxl
910:   c2c9 85 bd            sta  dsum
920:   c2cb a5 96            lda  dsumh
920:   c2cd 65 b7            adc  dxh
920:   c2cf 85 96            sta  dsumh
925:   c2d1 60               rts
930:                 ;---------------------------------
1000:                ;dsum = dsum - dx
1010:  c2d2 38       dsdxsb  sec

1010:  c2d3 a5 bd            lda  dsum
1010:  c2d5 e5 b6            sbc  dxl
1010:  c2d7 85 bd            sta  dsum
1020:  c2d9 a5 96            lda  dsumh
1020:  c2db e5 b7            sbc  dxh
1020:  c2dd 85 96            sta  dsumh
1030:  c2df 60               rts
1040:                ;---------------------------------
1100:                ;dsum = dsum - dy

1110:  c2e0 38       dsdysb  sec
1110:  c2e1 a5 bd            lda  dsum
1110:  c2e3 e5 b5            sbc  dy
1110:  c2e5 85 bd            sta  dsum
1110:  c2e7 b0 02            bcs  dsdysb1
1120:  c2e9 c6 96            dec  dsumh
1139:  c2eb 60       dsdysb1 rts
```

```
profi-ass 64 v2.0    seite 4

1190:                ;
1200:                ;--------------------------------
1205:                ;schritt in y-richtung
1210:  c2ec a5 95    ystep   lda  gy
1210:  c2ee d0 04            bne  ystep1
1220:  c2f0 20 8a c3         jsr  unten
1220:  c2f3 60               rts
1230:  c2f4 20 82 c3 ystep1  jsr  oben
1230:  c2f7 60               rts
1390:                ;
1400:                ;--------------------------------

1410:                ;schritt in x-richtung
1420:  c2f8 a5 94    xstep   lda  gx
1420:  c2fa d0 04            bne  xstep1
1430:  c2fc 20 62 c3         jsr  rechts
1430:  c2ff 60               rts
1440:  c300 20 6e c3 xstep1  jsr  links
1440:  c303 60               rts
1500:                ;--------------------------------
1510:                ; sw = sw -1

1520:  c304 38       sw1     sec
1520:  c305 a5 be            lda  swl
1520:  c307 e9 01            sbc  #01
1520:  c309 85 be            sta  swl
1530:  c30b b0 02            bcs  sw2
1530:  c30d c6 bf            dec  swh
1540:  c30f 60       sw2     rts
1550:                ;
1600:                ;--------------------------------
1610:                ; dsum >= dx

1620:  c310 38       dsdxcp  sec
1620:  c311 a5 bd            lda  dsum
1620:  c313 e5 b6            sbc  dxl
1630:  c315 a5 96            lda  dsumh
1630:  c317 e5 b7            sbc  dxh
1640:  c319 60               rts
1650:                ;
1700:                ;--------------------------------
1710:                ; dsum >= dy

1720:  c31a 38       dsdycp  sec
1720:  c31b a5 bd            lda  dsum
1720:  c31d e5 b5            sbc  dy
1730:  c31f a5 96            lda  dsumh
1730:  c321 e9 00            sbc  #00
1740:  c323 60               rts
1750:                ;
1800:                ;--------------------------------
39990:               ;--------------------------------
39995:               ; zeitschleife

40000: c324 a5 b4    delay   lda  time
40000: c326 f0 15            beq  delout
40010: c328 8d 94 c3         sta  zaehl1
40010: c32b 8d 95 c3         sta  zaehl2
40030: c32e ce 94 c3 del     dec  zaehl1
40040: c331 d0 fb            bne  del
40050: c333 a9 2a            lda  #$2a
40060: c335 8d 94 c3         sta  zaehl1
40065: c338 ce 95 c3 de2     dec  zaehl2
```

```
profi-ass 64 v2.0    seite 5

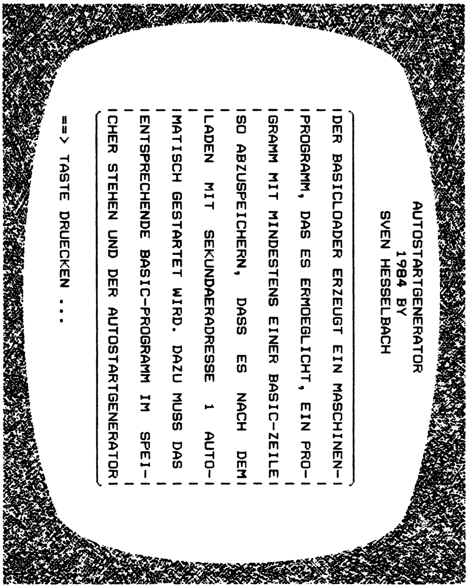
40066: c33b d0 f1               bne  del
40070: c33d 60          delout  rts
40990:                  ;--------------------------------
40995:                  ; neg. xdiff umkehren
41000: c33e a5 b6       xdinv   lda  dxl
41000: c340 49 ff               eor  #$ff
41000: c342 85 b6               sta  dxl
41010: c344 a5 b7               lda  dxh
41010: c346 49 ff               eor  #$ff
41010: c348 85 b7               sta  dxh
41015: c34a e6 b6               inc  dxl
41015: c34c d0 02               bne  xdie
41017: c34e e6 b7               inc  dxh
41020: c350 a9 ff       xdie    lda  #$ff
41020: c352 85 94               sta  gx
41020: c354 60                  rts
41900:                  ;--------------------------------
41995:                  ; neg. ydiff umkehren
42000: c355 a5 b5       ydinv   lda  dy
42000: c357 49 ff               eor  #$ff
42000: c359 85 b5               sta  dy
42000: c35b e6 b5               inc  dy
42000: c35d a9 ff               lda  #$ff
42000: c35f 85 95               sta  gy
42000: c361 60                  rts
42020:                  ;
42030:                  ;--------------------------------
43000: c362 e6 14       rechts  inc  xal
43010: c364 d0 02               bne  rel
43010: c366 e6 15               inc  xah
43030: c368 a6 a6       rel     ldx  y1
43030: c36a 20 fd c0            jsr  mm+6     ; mobsetzen
43070: c36d 60                  rts
```

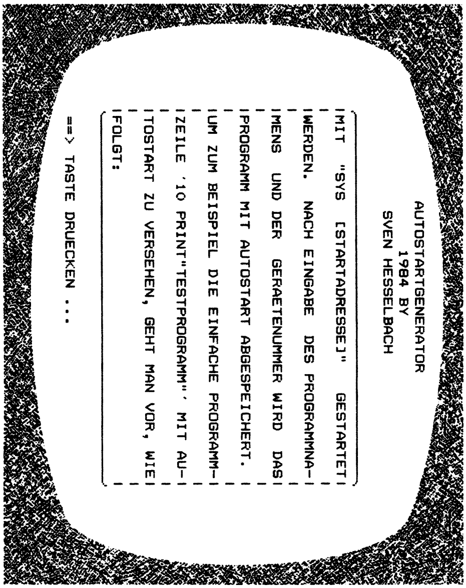
```
43090:                  ;--------------------------------
43100: c36e a5 14       links   lda  xal
43100: c370 08                  php
43100: c371 c6 14               dec  xal
43100: c373 28                  plp
43100: c374 d0 06               bne  li1
43110: c376 a5 15               lda  xah
43110: c378 f0 02               beq  li1
43110: c37a c6 15               dec  xah
43120: c37c a6 a6       li1     ldx  y1
43120: c37e 20 fd c0            jsr  mm+6
43140: c381 60                  rts
43190:                  ;--------------------------------
43200: c382 c6 a6       oben    dec  y1
43200: c384 a6 a6               ldx  y1
43200: c386 20 fd c0            jsr  mm+6
43210: c389 60                  rts
43290:                  ;--------------------------------
43300: c38a e6 a6       unten   inc  y1
43300: c38c a6 a6               ldx  y1
43300: c38e 20 fd c0            jsr  mm+6
43310: c391 60                  rts
43390:                  ;--------------------------------
50070: c392 4a 53               .asc "js"
50080: c394 00          zaehl1  .byte00

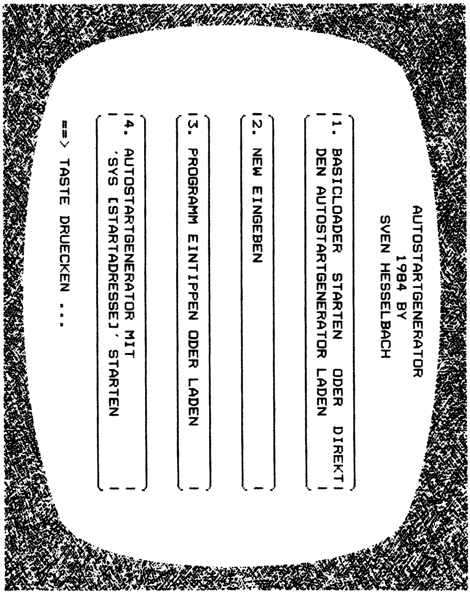

profi-ass 64 v2.0    seite 6

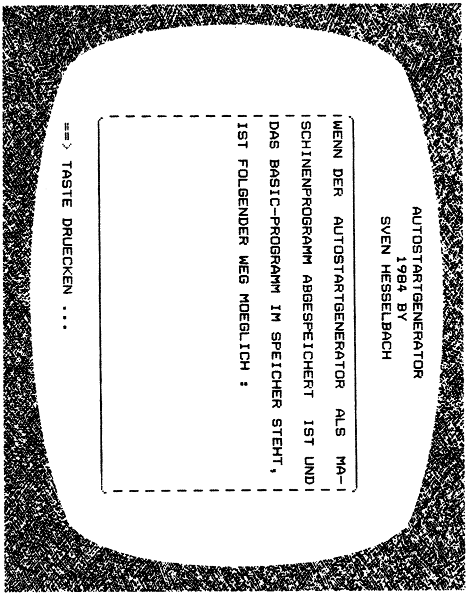
50090: c395 00          zaehl2  .byte00
50099: c396 00          zaehl3  .byte00
53000:                  ;--------------------------------
]c1e6-c397
no errors
```

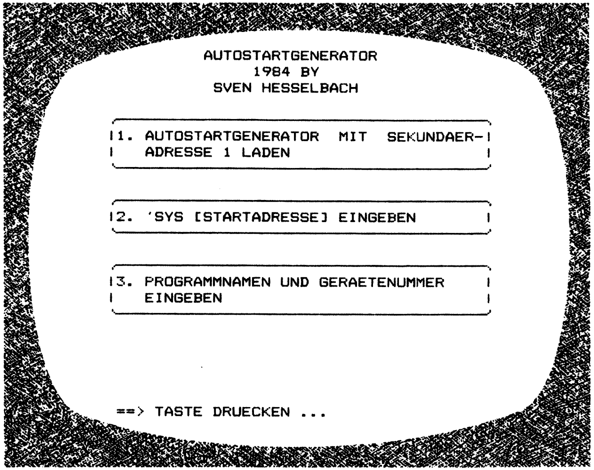
HARDCOPYS: "AUTOSTART"

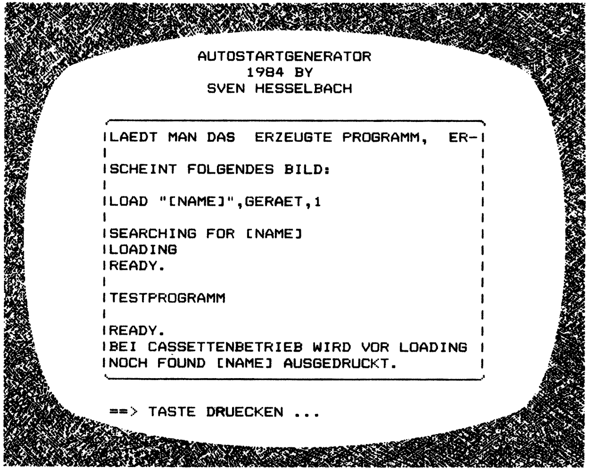












PS: IM KASSETTENBETRIEB

GENUEGT ZUM LADEN UND STARTEN EINES

AUTOSTART-PROGRAMMES EIN EINFACHES

LOAD "[NAME]"

ASSEMBLER-LISTING: "AUTOSTART"

```
]
0010          .OS
0020          .BA $C000
0030AUTOPGM   .DE 2024         ;LAGE DES STARTPROGRAMMS.
0040SAVESTART .DE 792          ;ANFANG DES ZU SPEICHERNDEN BEREICHS
0050          JSR $A533        ;BASIC-ZEILEN NEU BINDEN,
0060          CLC              ;UM DIE ZEIGER FUER BASIC-PROGRAMME
0070          LDA $22          ;NACH DEM NACHLADEN WIEDER IN ORDNUNG
0080          ADC #$02         ;ZU BRINGEN
0090          STA $2D
0100          STA $2F
0110          STA $31
0120          LDA $23
0130          ADC #$00
0140          STA $2E
0150          STA $30
0160          STA $32
0170          CLC
0180          ;-----------------------
0190          LDX #0           ;AUSGABE DES TEXTES,
0200DRUCK     LDA TEXT,X       ;DER AB 'TEXT' STEHT (S.U.)
0210          BEQ END1
0220          JSR $FFD2
0230          INX
0240          BNE DRUCK
0250          ;----------
0260END1      LDX #0           ;LAENGE DES NAMENS AUF 0
0270          STX 183
0280GETNAM    JSR $FFCF        ;SCHLEIFE ZUM EINLESEN DES FILENAMENS
0290          CMP #$0D
0300          BEQ END2
0310          LDX 183
0320          STA NAME,X
0330          INC 183          ;UM 1 ERHOEHEN
0340          CPX #18          ;LAENGER ALS 18?
0350          BCC GETNAM       ;NEIN, WEITER
0360          ;----------
0370END2      LDX #0
0380DRUCK1    LDA TEXT1,X      ;DRUCK DES TEXTES
0390          BEQ TAST         ;AB 'TEXT1'
0400          JSR $FFD2
0410          INX
0420          BNE DRUCK1
0430TAST      JSR $FF9F
0440          JSR $FFE4        ;ZEICHEN IM AKKU
0450          CMP #$31         ;NUR ZIFFERN ZULASSEN
0460          BCC TAST
0470          CMP #$3A
0480          BCS TAST
0490          SEC
0500          SBC #$30         ;ASCII IN WERT
0510          CLC
0520          STA $BA          ;UND ABSPEICHERN
0530          LDX #L,NAME      ;ZEIGER AUF FILENAMEN
0540          LDY #H,NAME
0550          LDA 183          ;LAENGE DES FILENAMENS
0560          JSR $FFBD        ;PARAMETER SETZEN
0570          LDX $BA          ;GERAETENUMMER
0580          LDY #1           ;SEKUNDAERADRESSE
0590          JSR $FFBA        ;SETZEN
0600          LDA #H,AUTOPGM   ;ANFANG DES STARTPROGRAMMS
0610          LDX #L,AUTOPGM
0620          STX 804          ;IN INPUT-VEKTOR
0630          STA 805
0640          JSR $E544        ;CLR/HOME
0650          ;----------
0660          LDX #0           ;SCHLEIFE, UM DAS MASCHINENPROGRAMM,
0670WRTPGM    LDA MPGM,X       ;DAS DAS BASICPROGRAMM NACH DEM LADEN
0680          STA AUTOPGM,X    ;STARTET, IN EINEN BEREICH (2024),
0690          CPX #MPGME-MPGM  ;DER MITGELADEN WIRD, ZU SCHREIBEN
0700          BEQ END4
0710          INX
0720          BNE WRTPGM
0730          ;----------
0740END4      LDA #0           ;PROGRAMMODUS, UM MELDUNG 'SAVING ...'
0750          STA 157          ;ZU UNTERDRUECKEN
0760          LDA #H,SAVESTART ;START DES ZU SPEICHERNDEN BEREICHS
0770          LDX #L,SAVESTART
0780          STA $FC          ;FESTLEGEN UND IN 251/252 ABLEGEN
0790          STX $FB
0800          LDA #$FB         ;AKKU MIT 251 LADEN
0810          LDX $2D          ;X/Y-REGISTER LEGEN PROGRAMMENDE FEST
0820          LDY $2E          ;(45/46) = VARIABLENSTART+PROGRAMMENDE
0830          JSR $FFD8        ;'SAVE'
0840          LDA #$F1         ;INPUT-VEKTOR WIEDER AUF ALTEN WERT
0850          LDX #$57         ;UMLEGEN ($F157)
0860          STA 805
0870          STX 804
0880          RTS
0890          ;-----------------------
0900TEXT      .BY $93 '********** AUTOSTARTGENERATOR **********' 13
0910          .BY '****** (C) 1984 BY SVEN HESSELBACH *****' 13
0920          .BY 'PROGRAMMNAMEN EINGEBEN: ' 0
0930          ;----------
0940TEXT1     .BY 13 13 'GERAETEADRESSE (1-9) ? ' 0
0950NAME      .DS 19           ;FUER NAMEN 19 BYTES RESERVIEREN
0960          ;----------
0970          ;---------- ES FOLGT DAS PROGRAMM, DAS DAS BASIC-PROGRAMM
0980          ;---------- STARTET. ES STEHT IN SPRITEBLOCK 11
0990MPGM      LDA #$F1         ;INPUT-VEKTOR WIEDER AUF ALTEN WERT
1000          STA 805          ;LEGEN, DA ER BEIM LADEN VERAENDERT WURDE
1010          LDA #$57
1020          STA 804
1030          JSR $A68E        ;PROGRAMMZEIGER AUF BASICSTART LEGEN
1040          LDX #0
1050          JSR $A871        ;'RUN' AUSFUEHREN
1060          JMP $A7AE        ;IN INTERPRETERSCHLEIFE SPRINGEN
1070MPGME     .BY 0            ;ENDE DES MASCHINENPROGRAMMS
1080          .EN
//
]
```

F O R M E L P L O T T E R

Nach dem Starten durch 'RUN' erscheint ein Hauptmenue mit den Anweisungen: -FUNKTION DEFINIEREN, -WERTE AUSGEBEN, -FUNKTION PLOTTEN.

Nachdem eine Funktion definiert wurde, können die Werte ausgegeben, oder die Funktion geplottet werden.

WERTE AUSGEBEN:

Hier können Sie den Bereich fuer 'X' definieren (zum Beispiel: -10 bis 30) und die Schrittweite eingeben. Danach werden die Werte einzeln auf dem Bildschirm angezeigt.

PLOTTEN:

Im PLottmodus können Sie durch J/N wählen, ob alle Koordinaten vordefiniert werden sollen oder nicht.
Koordinaten definieren: hierbei muß beachtet werden, daß für 'X MIN' ein kleinerer Wert als für 'X MAX' eingegeben werden muss. Genauso muss auch bei 'Y MIN' und 'Y MAX' verfahren werden.
Viel Spass beim Plotten!

Wichtig: SAVEn nur zu Anfang möglich, nachdem die Grafik einmal eingeschaltet war nicht mehr!

M L M 6 4 - E I N K O M F O R T A B L E R M O N I T O R

Der Autor des Programms, Pascal Dornier, Zollikon, Schweiz, erläutert im folgenden Text die Bedienung des Maschinensprachemonitors.

S Y N T A X

- Es gibt keinen Prompt.
- Zur Trennung von Parametern und Daten wird konsequent das Leer zeichen verwendet. Mehrere Leerzeichen werden toleriert.

- Zahlen können abgekürzt werden, zum Beispiel kann $08 als 8 eingegeben werden.

- Bei Fehleingabe erscheint ein inverses Fragezeichen und gibt einen sanften Fingerzeig, wo der Fehler liegt.

- Zahlen können folgendermassen eingegeben werden:

```
  $41     Hex
   65     Dezimal
  %01100101 Binär
' A       ASCII
```

Alle Daten können Hexadezimal, Dezimal oder Binär ausgegeben werden, je nachdem, welches Zahlensystem eingeschaltet ist (Beim Programmstart Hex.). Die Zahlen dieser bevozugten Basis können ohne Präfix ($/ /%) eingegeben werden. Im Binär-Modus werden Adressen hexadezimal ausgegeben, 16-Bit-Binärzahlen können nicht eingegeben werden.

-Alle Ausgaben können mit der CTRL-Taste verlangsamt, mit SPACE angehalten und mit COMMODORE wieder weitergeführt werden. Mit der STOP-Taste können die Ausgaben abgebrochen werden.

Zur Beschreibung der Syntax werden folgende Abkürzungen verwendet:

a =Anfangsadresse
e =Endadresse
z =Zieladresse
b =Byte
/ =von hier an sind alle Parameter fakultativ

G / a
Das Maschinenprogramm bei a oder beim Stand des Programmzählers wird durch diesen Befehl gestartet, wobei alle Register geladen werden und die Ausführung durch den Maschinenbefehl BRK abgebrochen wird.

H a e b ...
Durch den Befehl H(unt) wird das Byte/ werden die Bytes b im Speicherbereich von a bis e und alle Fundstellen angezeigt.
Beispiel: H C000 CBCF D2 sucht im Monitor alle Aufrufe der Ausgaberoutine.

L "Name" b / z
Dieser Befehl lädt ein Programm von Kassette (b=1) oder Diskette (b=8) in den Speicher, wobei durch Eingabe von z eine andere Zieladresse vorgegeben werden kann.

M / a / e
Der Speicherbereich von a bis e wird als Dump ausgegeben. Die Varianten der Syntax sind gleich wie bei D. Vor den Befehl M können verschiedene Präfixe gestellt werden: + bewirkt die Ausgabe von Prüfsummen.
Beispiel: +PM C000 CBCF gibt ein Hex-Dump des Monitors mit Prüfsummen auf den Drucker aus.
bewirkt die Ausgabe eines ASCII-Dumps.

P alle Befehle
Dieses Präfix kann vor alle Befehle gestellt werden und leitet alle Ausgaben des darauffolgenden Befehls auf File 4. Falls dieser File noch nicht vorhanden ist, wird OPEN 4,4 ausgeführt, die Ausgabe also auf den Drucker geleitet. Man kann vor dem Start des Monitorprogramms ein File eröffnen (z.B. OPEN4,8,2,"Beispiel,S,W") und so ein Disassemblerlisting auf Diskette schreiben. Nach dem Verlassen des Monitors muß der File geschlossen werden. Ein nützliches Bug: P allein bewirkt die Ausgabe eines Zeilenvorschubs auf den Drucker.

Q / a
Dieser Befehl funktioniert gleich wie E, gibt aber kein Protokoll aus, so daß man die Ausgaben oder Grafik des getesteten Programms sehen kann. Bei einem Abbruch wird die "History" ausgegeben, ein Trace für die vier zuletzt ausgeführten Befehle. Die Programmausführung ist bei Q etwa 130 mal langsamer als normal, es ist also nicht empfehlenswert, mit dieser Routine Verzögerungsschleifen zu testen.

B E F E H L E

Die Funktionen der verschiedenen Befehle werden nun eingehend erläutert.

A a
Hexdump wird ignoriert. Mnemonic Parameter Beispiel:

```
A 0800 4C 00 00 :JMP 0000
```

Mit diesem Befehl können Maschinenprogramme im 6502-Mnemonicformat eingegeben werden. Nach Eingabe einer Assemblerzeile bleibt der Monitor im Assembler-Modus, bis eine fehlerhafte Zeile eingegeben wird. Der Assembler gibt auf jeder Zeile die laufende Adresse aus und setzt den Cursor auf den Beginn des Mnemonic-Feldes, das nach dem Doppelpunkt beginnt. Mit dem Assembler können von D disassemblierte Programme editiert werden. Dabei ist auch die Verschiebung von kürzeren Programmstücken möglich, wobei die Sprungadressen von Hand korrigiert werden müssen.

B / b a
Beispiel: B 3 4000
Der Befehl B ohne Parameter zeigt die Breakpoints an. Das Beispiel setzt den Breakpoint 3 auf die Adresse $4000. Es gibt die Breakpoints 0 bis 3. Es handelt sich um "weiche" Breakpoints, die nur von der Trace-Routine (E,Q) erkannt werden.

C a e z
Beispiel: C 1000 107F 4096
Der Befehl C vergleicht den Speicherbereich von a bis e mit dem bei z beginnenden Speicherbereich. Alle Unterschiede werden angezeigt. Eine mögliche Anwendung ist die Suche nach

Änderungen, falls man sich selber nicht mehr in seinen Programmversionen auskennt.

D / a / e
Beispiele: D C000 C07F disassembliert das Monitorprogramm von $C000 bis $C07F.

D C080 disassembliert 22 weitere Zeilen.

E / a
Dieser Befehl ermöglicht den Einzelschrittbetrieb des Prozessors. Das bei a oder dem laufenden Programmzählerstand stehende Programm wird ausgeführt, wobei jeder ausgeführte Befehl disassembliert wird. Das Programm hält an, wenn es auf einen undefinierten Befehl oder auf einen Breakpoint stösst.

F a e b
Beispiel: F 0800 9FFF EA
Der Speicherbereich von a bis e wird mit dem Byte b gefüllt (z.B. NOP's).

R
Dieser Befehl zeigt die Register an. Der Prozessorstatus P wird immer binär ausgegeben. Die Daten können auf dem Bildschirm geändert und durch Drücken von Return, also durch Ausführung des Befehls *, neu gesetzt werden.

S "Name" b a e
Der Befehl S speichert den Speicherbereich von a bis e auf Kassette (b=1) oder Diskette (b=8).

T a e z
Dieser Befehl überträgt den Speicherbereich von a bis e zum Speicherbereich, der bei z beginnt. Eine Adressenumrechnung erfolgt nicht.

V "Name" b / z
Dieser Befehl entspricht L, führt aber statt "Load" ein "Verify" durch.

X
Dieser Befehl bewirkt die Rückkehr zu Basic.

Y / a
Dieser Befehl funktioniert gleich wie G, setzt aber den Stackpointer nicht, so daß auch eine Rückkehr mit RTS möglich ist.
Beispiel: Y C020 gibt Return aus.

&, ,% Daten
Mit diesen Befehlen kann man das gewünschte Zahlensystem setzen. Eventuell von M ausgegebene Daten werden in den Speicher geschrieben, falls vorhanden wird die Prüfsumme getestet.

D A S P R O G R A M M

Die Dezimal-Umwandlungsroutine stammt aus (1). Die Syntaxprüfung ist programmtechnisch interessant gelöst, und zwar mit einem "Syntax-Interpreter", wie man ihn im ROM des ZX-81 findet (2).

Dieser sucht den Befehl in der Syntaxtabelle, liest die benötigten Daten ein, prüft die Syntax und springt erst dann zum eigentlich Befehl.

Bei genügender Kenntnis der Innereien des Programms kann man es auch für andere Zwecke missbrauchen, z.B. einen Disassembler für einen anderen Prozessor aufpfropfen. Das Monitorprogramm belegt den Speicher von $C000 bis $CBCF (bzw. $9000 bis $9bcf), außerdem die Zero-Page-Adressen von $55 bis $5A, von $5E bis $67, von $69 bis $6F und von $FB bis $FE.

B E N U T Z E R H I N W E I S E

Sie erhalten den Monitor gleich zweimal: für den Adressbereich $C000 - $CBCF und $9000 - $9BCF. Nach dem Starten des vom eigenen Datenträger geladenen Programms erfolgt ein Sprung in den oberen Monitor.

Der Monitor kann sich mit

```
S "MLMC" 8 C000CBF
```

selbst auf Diskette abspeichern, den zweiten mit

```
S "MLM9" 8 $9000-$9BCCF
```

(bzw. Geräteadresse "1" für Kassette oder "7" für SuperTape)
Das Programm kann nun direkt von Basic aus mit LOAD "MLM",8,1 geladen werden. Danach sollte man NEW ausführen, damit die Basic-Zeiger wieder in Ordnung sind. Der Start erfolgt mit:

```
SYS 49847 bzw. SYS 37559.
```

P.D.

(Das 22 Seiten lange Source-Listing sprengt den Rahmen dieses Beiheftes. Es ist beim Verlag gegen 10 DM (Fotokopien, Verpackung, Porto)erhältlich.)

L i t e r a t u r

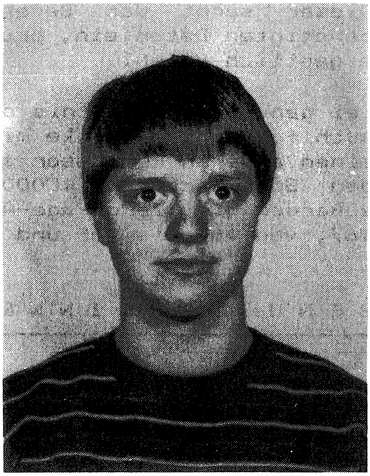
(1) Laher, Friedrich, SED hilft bei Binär-Dezimal-Umwandlung, mc 3/81
(2) Dornier, Pascal, ZX-81-ROM-Listing, Eigenverlag.

3 0 0 0 M A R K G E W O N N E N

Der erste Sieger im INPUT 64-Wettbewerb steht fest:

Wolfgang Dunczewski mit seinem Spiel "HELLS BELLS". Die Redaktion war sich in der Beurteilung einig. Die grundlegende Spielidee ist zwar nicht neu, doch der Autor hat viele Ideen in die kleinen, aber entscheidenden Feinheiten gesteckt. Und: Das Spiel ist "professionell" programmiert - das geht anscheinend auch schon mit siebzehn Jahren.

Der Autor über sich selbst:



Mein Name ist Wolfgang Dunczewski. Ich wurde am 5. März 1967 geboren und besuche die 11.Klasse eines Oberstufengymnasiums.

Weihnachten 1982 bekam ich einen VC 20 geschenkt. Mit Datasette kostete er damals fast 800.- DM. Schon in den Weihnachtsferien begann ich, kleine Programme in Basic zu schreiben. Etwas später fing ich damit an, mit Hilfe von Büchern Assembler anzuwenden. Ende 1983 trat ich meinen VC 20 an meinen 6 Jahre jüngeren Bruder ab und schaffte mir einen C 64 an. Da ich einige Erfahrung vom VC 20 hatte, gelang es mir schon bald - mit Hilfe der Sprites - Spielprogramme zu schreiben. An meinem Spiel "HELLS BELLS" habe ich vier Wochen gearbeitet.

Wolfgang Dunczewski, Frankfurt

D E R W E T T B E W E R B G E H T W E I T E R !!

Die nächsten 3000 Mark warten auf einen neuen Gewinner. Noch einmal kurz die Bedingungen für den INPUT 64-Wettbewerb:

Sie können Grafikprogramme, Musikprogramme, Spiele, Lernprogramme, Anwenderprogramme und völlig neue Programmideen einsenden.

Wichtig: Werfen Sie einen Blick in das Kapitel "Technisches", damit Ihr Programm auch innerhalb von INPUT 64 lauffähig ist.

Der Rechtsweg ist wie immer ausgeschlossen.

BEI LADEPROBLEMEN:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft.

Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben.

Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten sehr häufig auf.
Wir haben ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme-/Wiedergabekopf justieren können.

Tippen Sie das Programm JUSTAGE ein und speichern Sie es ab. Dieses Programm wertet ein cirka 30 Sekunden langes Synchronisationssignal aus, das sich am Ende des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen einen Blick auf Ihre Datasette. über der REWIND Taste, in etwa 0,5 cm Abstand vom Kassettenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage.

Legen Sie nun die zweite Seite von INPUT64 ein und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

Starten Sie das JUSTAGE-Programm mit .RUN., jetzt sollte die Meldung PRESS PLAY ON TAPE kommen, drücken Sie die PLAY Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchrosignal erreicht, wechselt die Bildschirmfarbe; und zwar - bei nicht total verstellter Spurlage - völlig regelmäßig etwa drei mal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig. Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber Vorsicht: ganz langsam drehen ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als 1 Umdrehung in jede Richtung. Nach etwas

Ausprobieren,wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

Zur Feineinstellung lassen Sie das Synchro Signal noch einmal von Anfang laufen. Die Schraube jetzt nach links drehen , bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken (am Besten markieren) und die Schraube jetztlangsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei

weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, d.h. zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf!

Der Tonkopf Ihres Recorders ist jetzt exakt justiert. Sollte sich auch nach dieser Einstellung INPUT64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie dazu bitte die defekte Kassette mit einem entsprechenen Vermerk an den Verlag (Adresse siehe Impressum) ein.

Listing JUSTAGE

```
800 fori=49199to49410:read d:ps = ps + d:poke i, d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys49338
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217,174,  4,220,172,  5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144,  4,224,115,176,  7,169,  0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88,201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165,  1, 41, 31,133,  1,133,192,169, 47
1100 data141, 20,  3,169,192,141, 21,  3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141,  4,220,169,129,141
1120 data  5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76

ready.
```

B E R I C H T I G U N G E N

SUPERTAPE/D (INPUT64 2/85)

Die SuperTape-Version aus dieser Ausgabe ist nicht ohne weiteres als Basic-File lauffähig. Sondern: Laden Sie SuperTape von Ihrem eigenen Datenträger zunächst ganz normal. Laden Sie es dann noch einmal mit der Sekundäradresse 1 nach (LOAD "SUPERTAPE",8,1 bzw. ...,1,1). Saven Sie die so entstandene Version erneut ab.

DATEIKASTEN (INPUT64 1/85)

Leider weist das Programm zwei Fehler auf. Wir bitten um Entschuldigung und stellen ein kurzes Patchprogramm zu Verfügung, daß diese behebt. Eventuell 'verlorengegangene' Dateien lassen sich mit dem gepatchten Programm wieder einlesen. Leider wurden einige falsche Berichtigungen verschickt, die richtige Version enthält in der ersten Zeile die Kennzeichnung (V2).

-Listing abtippen und mit .RUN. starten
- -DATEIKASTEN von Ihrem Dateiträger, nicht aus dem Magazin laden.
- -.SYS 49 207. und .RETURN. eingeben.
- -Die jetzt korrigierte Version des Programms .SAVE.n.

```
1000 Rem Berichtigung Dateikasten (V2)
1010 :
1020 for i=49152 to 49236 : read d : ps=ps+d : poke i,d : next
1030 if ps  11132 then print" *** Tippfehler ***" : end
1040 new
1050 :
1060 data 160, 43,185, 12,192,153, 57, 39,136, 16,247, 96, 32
1070 data 207,255,133,247, 32,207,255,133,248,169,  0,133,141
1080 data 169, 56,133,142,165,247,197,141,208,  6,165,248,197
1090 data 142,240, 25, 32,207,255,160,  0,145,141, 32,115, 39
1100 data  76, 75, 39,160,  0,185, 71,192,153, 78, 37,200
1110 data 192, 14,208,245, 76,  0,192,165,167, 24,105,  2,133
1120 data 201,165,168,105,  0,133,202,234
```

SOUNDCONTROL (INPUT64 1/85)

Zu unserem Bedauern ist das Programm SOUNDCONTROL nicht wie versprochen .SAVE.bar. Um es dennoch zu .SAVE.n gehen Sie bitte folgendermaßen vor:

1. Laden Sie innerhalb von INPUT64 SOUNDCONTROL.
2. Drücken Sie gleich zu Anfang .RUN/STOP. und .RESTORE.
3. Geben Sie im Direktmodus ein: POKE 45,4 : POKE 46,100 : POKE 56,160 : CLR .RETURN.
4. Jetzt können Sie das Programm mit SAVE "name",gerätenummer abspeichern.
5. Zurück ins INPUT64 Betriebssystem mit SYS 50307.

BITS & BYTES IM VIDEOCHIP Teil II (INPUT64 2/85)

Auf Seite 12 des Kurses ist das Ergebnis der OR Verknüpfung natürlich falsch, wie aufmerksame Kursteilnehmer sicherlich schon bemerkt haben. Als Ergebnis wird die zweite Verknüpfungszahl ausgegeben, wie es richtig sein muß, ergibt sich aus der Seite 11.

Ab 15. April an Ihrem Kiosk:

I N P U T 6 4 Ausgabe 4/85

Wir bringen unter anderem:

* Grafik-Paket
  Schnelle Assemblerroutinen zur Bildschirmzauberei

* Short-Save
  Ein Tool zur Speicherplatzersparnis

* Video-Chip-Kurs über hochauflösende Grafik, Spiele,
  Anwenderprogramme, News, 64er-Tips zu Stringoperationen

IMPRESSUM

INPUT 64
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 2746
3000 Hannover 1
Tel.: (05 11) 53 52-0

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Ständige Mitarbeiter:
Peter Berk
Peter Sager
Hajo Schulz
Eckart Steffens

Vertrieb: Anita Kreutzer

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:

Verlag Heinz Heise GmbH
Postfach 2746
3000 Hannover 1
Tel.: (05 11) 53 52-0

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:
Köhler & Lippmann, Braunschweig.

Druck:
Leunisman GmbH, Hannover
Hahn-Druckerei, Hannover

Konfektionierung:
Lettershop Brendler, Hannover

Kassettenherstellung:
Visoton Cassettentechnik, Bonn

INPUT 64 erscheint monatlich.
Einzelpreis DM 12,80
Jahresabonnement Inland DM 140,-

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61

Eine Verantwortung für die Richtigkeit der Veröffentlichungen und die Lauffähigkeit der Programme kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

ISSN 0177-3771

Titelidee: **INPUT 64**
Titelfoto: Foord u. Schumacher/Artreferenz

hier abtrennen

INPUT 64-Abonnement Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat

____

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.

○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift ______________________________

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

INPUT 64-Abonnement Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl Konto-Nr.

Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Heise

Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT 64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1